No-substitution k-means Clustering with Adversarial Order

Robi Bhattacharjee RCBHATTA@ENG.UCSD.EDU
University of California, San Diego

Michal Moshkovitz MMOSHKOVITZ@ENG.UCSD.EDU
University of California, San Diego

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We investigate k-means clustering in the online no-substitution setting when the input arrives in *arbitrary* order. In this setting, points arrive one after another, and the algorithm is required to instantly decide whether to take the current point as a center before observing the next point. Decisions are irrevocable. The goal is to minimize both the number of centers and the k-means cost. Previous works in this setting assume that the input's order is random, or that the input's aspect ratio is bounded. It is known that if the order is arbitrary and there is no assumption on the input, then any algorithm must take all points as centers. Moreover, assuming a bounded aspect ratio is too restrictive — it does not include natural input generated from mixture models.

We introduce a new complexity measure that quantifies the difficulty of clustering a dataset arriving in arbitrary order. We design a new random algorithm and prove that if applied on data with complexity d, the algorithm takes $O(d\log(n)k\log(k))$ centers and is an $O(k^3)$-approximation. We also prove that if the data is sampled from a "natural" distribution, such as a mixture of k Gaussians, then the new complexity measure is equal to $O(k^2\log(n))$. This implies that for data generated from those distributions, our new algorithm takes only $\mathrm{poly}(k\log(n))$ centers and is a $\mathrm{poly}(k)$-approximation. In terms of negative results, we prove that the number of centers needed to achieve an α-approximation is at least $\Omega\left(\frac{d}{k\log(n\alpha)}\right)$.

Keywords: k-means clustering, online no-substitution setting, adversarial order, complexity measure, the online center measure, mixture models

1. Introduction

Clustering is a fundamental task in unsupervised learning with many diverse applications such as health Zheng et al. (2014), fraud-detection Sabau (2012), and recommendation systems Logan (2004) among others. The goal of k-means clustering is to find k centers that minimize the k-means cost of a given set of points. The cost is the sum of squared distances between a point and its closest center. This problem is NP-hard Aloise et al. (2009); Dasgupta (2008), and, consequently, approximated algorithms are used Arthur and Vassilvitskii (2006); Aggarwal et al. (2009); Kanungo et al. (2004). An algorithm is an α-approximation if the k-means cost of its output is at most α times the optimal one.

In this paper, we focus on the online no-substitution setting where the points arrive one after another, and decisions are made instantly, before observing the next point. In this online setting, the algorithm decides whether to take a point as center immediately upon its arrival. Decisions cannot be changed; once a point is considered a center, it remains center forever. Conversely, points that are not centers cannot become centers after the next point is received. This set-up is summarized in Algorithm 1. In this setting, the goal is two-fold: (i) minimizing the cost of the returned centers,

Algorithm 1: Online no-substitution setting

$C \leftarrow \{\}$; // set of centers

for $t \in \{1, 2, 3, \ldots, n\}$ **do**

  decide whether to add x_t to C; // only x_t can be added to C at time t, and no center can be removed from C

end

return C

C, and (ii) minimizing number of centers, $|C|$.

The importance of the online no-substitution setting is motivated by several examples in Hess and Sabato (2020). Here is one of them: suppose we are running a clinical trial for testing a new drug. Patients come one after another to the clinic, and for each of them we must decide whether or not to administer the drug. Our choices are immutable: once a patient takes the drug, it cannot be untaken, and once a patient leaves the clinic, we cannot decide to test the drug on them. Our overall goal is to administer the drug to a small representative sample of the entire population. In this example, the patients are the points, and the people given the drug are the centers. The number of patients that took the drug is the number of centers, which should be small as it is an experimental drug and thus risky. Assuming an appropriate distance measure between any two people, a low cost k-means clustering provides a good representation of the entire population. In this setting, previous works Hess and Sabato (2020); Moshkovitz (2019) present algorithms under the assumption that the order is random. However, this is not always the case. In the motivating example, the elderly patients might tend to arrive earlier than the younger ones. To account for this, in this paper we focus on the case that the order is arbitrary and might be adversarial.

It is known that if n input points arrive in arbitrary order, then $\Omega(n)$ centers are needed, as we demonstrate next. To prove this lower bound for any α-approximation algorithm, consider an exponential series of points in $\mathbb{R}$: $(2\alpha)^1, (2\alpha)^2, \ldots$. This sequence is constructed so that each point is very far away from the points preceding it. At each time step t, suppose that the algorithm doesn't select the current point, $(2\alpha)^t$. Because the number of points, n, is not known in advance, the algorithm must assume at all times that the series might stop. If this happens, the algorithm cost is roughly $(2\alpha)^t$, because of the cost incurred by the point $(2\alpha)^t$. On the other hand, the cost of the optimal clustering is roughly $(2\alpha)^{t-1}$, which violates the fact that the algorithm is an α-approximation. Therefore, the algorithm must select every point in the sequence. Observe that this lower-bound dataset is well-designed and pathological. This leads to the following questions: are there more "natural" datasets that require $\Omega(n)$ centers, and can we find a property that is shared by all hard-to-cluster datasets?

In this paper, we define a new measure for datasets that accurately captures the hardness of learning in the no-substitution setting with arbitrary order. We show that the lower-bound pathological example from above is the only reason for necessarily taking a large number of points as centers. We also design a new algorithm that takes a small number of centers whenever the new measure is small. Additionally, we prove that for data generated from many "natural" distributions (like Gaussian mixture models), the new measure is small ($O(k^2 \log n)$). The reason, intuitively, is that when sampling n points for many distributions, the largest subset with distances exponentially increasing is of length $O(\log n)$. This allows taking only $\mathrm{polylog}(n)$ centers. This number of centers is so small that it is close to the number of centers needed if the order is random Moshkovitz (2019).

1.1. Problem setting

Fix a dataset X with $|X| = n$ points. We define a clustering of X both by its clusters and its centers $\mathcal{C} = \{(C^1, c^1), (C^2, c^2), \ldots, (C^k, c^k)\}$, where $X = C_1 \sqcup \ldots \sqcup C_k$ is a partition of X, and c^i is the center of the cluster C^i. A clustering's k-means cost with distance metric[1] d is equal to

$$\mathcal{L}(\mathcal{C}) = \sum_{i=1}^{\ell} \sum_{x \in C^i} d(x, c^i)^2.$$

The optimal k-clustering, opt_k, with cost $cost(opt_k)$, is the one the minimizes the cost

$$opt_k = \underset{\substack{\mathcal{C}=((C^1,c^1),\ldots,(C^k,c^k)) \text{ s.t.} \\ X=C_1 \sqcup \ldots \sqcup C_k}}{\arg\min} \mathcal{L}(C).$$

Sometimes we define a clustering only by its centers or only by its clusters, and assume that the missing clusters or centers are the best ones. The best center for a cluster is its mean, and the best clusters for a given set of centers $c^1, \ldots, c^k$ are the closest points to each center $C^i = \{x : i = \arg\min_{j \in [k]} d(x, c^j)\}$.

The focus of the paper is the no-substitution setting where points arrive in an **arbitrary order**.

An online algorithm should return a clustering that is comparable to opt_k. But, for that to happen, the number of centers must be larger than k Moshkovitz (2019). Therefore, online algorithms return $\ell \geq k$ centers. We formally define "comparable to opt_k" in the following way. An α-approximation algorithm relative to opt_k is one that returns a clustering $\mathcal{C} = ((C^1, c^1), \ldots, (C^\ell, c^\ell))$ such that with probability[2] at least 0.9, $\mathcal{L}(\mathcal{C}) \leq \alpha \cdot \mathcal{L}(opt_k)$. The goal is to find an α-approximation algorithm that returns ℓ centers and aims to minimize *two* values: both α and ℓ. There is a trade-off between the two. Here are two extreme examples. If ℓ is large and equals n, then α is small and even equal to zero. If α can go to infinity, then ℓ can be small and equal to 1. In this paper, we focus on the case that $k \ll n$ or even k a constant. Thus, α should be independent of n and might be equal to $\mathrm{poly}(k)$.

1.2. Our results

In this paper, we design an algorithm that can learn in the no-substitution setting, even when points arrive in an *arbitrary* order. We showed that it is not possible for all datasets. Therefore, we introduce a new complexity measure that quantifies the hardness of learning a dataset in this setting. We show that the number of centers taken by the algorithm depends on the new complexity measure, and prove it is small for many datasets. We now formally define the new complexity measure.

An essential notion is a *diameter* of a set of points. For a set of points X its diameter, $\mathrm{diam}(X)$, is the distance of the two farthest points, $\mathrm{diam}(X) = \max_{x,x' \in X} d(x, x')$. The diameter of a clustering $\mathcal{C}$ is the maximal diameter among all of its clusters, $\mathrm{diam}(C) = \max_{C^i \in \mathcal{C}} \mathrm{diam}(C^i)$. The

1. d should be for all x, y (i) non negative $d(x, y) \geq 0$ (ii) $d(x, x) = 0$ and (iii) symmetric $d(x, y) = d(y, x)$
2. The probability is over the randomness of the algorithm; 0.9 can be replaced by any constant close to 1, this constant was used in Moshkovitz (2019).

best-ℓ-diameter (or ℓ-diameter for short) of a set of points X, $\text{diam}_\ell(X)$, is the diameter of the clustering with ℓ clusters and smallest diameter

$$\text{diam}_l(X) = \min_{\substack{\mathcal{C}=((C^1,c^1),\ldots,(C_\ell,c^\ell)) \text{ s.t.} \\ X=C_1 \sqcup \ldots \sqcup C_\ell}} \text{diam}(\mathcal{C}).$$

Earlier, we presented an exponential series that required taking $\Omega(n)$ centers. This lower-bound dataset required many centers because there is an order of the dataset where each new point seems to start a new cluster at its arrival. Intuitively, a new point x_t starts a new cluster if its distance to the closest point in $X_{t-1} = \{x_1, \ldots, x_{t-1}\}$ is farther than the diameter of the best clustering of X_{t-1} into $k-1$ clustering. More formally, the new complexity measure, Online Center measure, $\text{OC}_k(X)$, is defined as follows.

Definition 1 *The Online Center measure of a set X, denoted $\text{OC}_k(X)$, is the length of the largest sequence of points in X, $x_1, \ldots, x_m \in X$, such that for every $1 < j \leq m$,*

$$\min_{1 \leq i \leq j-1} d(x_i, x_j) > 2\text{diam}_{k-1}(\{x_1, x_2, \ldots, x_{j-1}\}).$$

The new measure, $\text{OC}_k(X)$, of a dataset X with $|X| = n$ and k clusters is an integer in $[n]$. The constant 2 can be replaced by any scalar $\alpha > 1$, see Lemma 7. As a sanity check, note that the lower-bound dataset has the highest OC possible, n, and indeed, the maximal number of centers is required for this dataset.

In the paper, we show that if the data is sampled from a "natural" mixture distribution then $\text{OC}_k(X) = O(k^2 \log n)$. Many distributions satisfy this:

Theorem 2 (informal) *With probability of about $1 - \frac{1}{n}$, sample X of n points from k mixture of: Gaussian, uniform, or exponential distributions has $\text{OC}_k(X) = O(k^2 \log n)$.*

One of the main contributions of this work is designing an algorithm in the no-substitution setting that uses a small number of centers, even if the points' order is adversarial. We prove that Algorithm 2 uses $O(\text{OC}_k(X) \log n)$ centers, when k is constant.

Theorem 3 (main theorem) *Let A be an α-approximation offline algorithm and X a set of points to be clustered. Algorithm 2 gets as an input X in the no-substitution setting in* arbitrary order *and returns a set of centers S with the following properties*

1. *with probability 0.9, the cost is bounded by*

$$cost(S) \leq 1358\alpha k^3 cost(opt_k)$$

2. *expected number of centers is bounded by*

$$\mathbb{E}[|S|] \leq 160\text{OC}_k(X)k\log(k)(\log n + 1).$$

The main theorem gives an upper bound on the expected number of centers taken by Algorithm 2 and its approximation quality. In particular, it takes $O(\text{OC}_k(X)k\log(k)\log(n))$ centers, and it is an $O(\alpha k^3)$-approximation. This means an exponential increase in the number of centers, without any information on the data. Note that the algorithm does not need to know what $\text{OC}_k(X)$ is.

Furthermore, if the data is generated from a distribution, the algorithm does not need to know this distribution, and it does not learn this distribution.

The idea of the algorithm is the following. For any optimal cluster C_i^*, it is well known that if we take each point in C_i^* with probability inversely proportional to $|C_i^*|$, then we get a good center for the entire cluster C_i^*. We cannot use this observation as-is because in the online setting, when a point x_t appears, we do not know the cluster C_i^* that it belongs to, and more importantly, we do not know its size, $|C_i^*|$. The algorithm tries to give an estimate for $|C_i^*|$. If this estimate is too small, the algorithm might take too many centers. If the estimate is too large, the algorithm might not take a good center from each optimal cluster. To find a reliable estimate, we use, as a subroutine, an approximated clustering algorithm A for the offline setting. After we observe a new point x_t we call A on the points observed so far, including x_t and get clustering C^t. We use the cluster that x_t is in, $x_t \in C_i^t$, to estimate the probability to take x_t. The value $|C_i^t|$ might be too small and unreliable. Therefore, the algorithm merges into C_i^t all clusters that are close to C_i^t. We show that after this merge, the estimation is just right, and we can bound both the approximation and the number of centers taken by the algorithm.

Another contribution of the paper is a proof that $\mathrm{OC}_k(X)$ lower bounds the number of centers taken by any algorithm in the online no-substitution setting. We prove that if $\mathrm{OC}_k(X)$ is large, many centers need to be taken by any algorithm.

Theorem 4 (lower bound) *Let X be an arbitrary set of points. There exists an ordering of X such that any α-approximation algorithm on X, the expected number of centers is $\Omega\left(\frac{OC_k(X)}{k\log(n\alpha)}\right)$.*

To illustrate our results, in Table 1, we summarize some of them for constant k. Each cell in the table states the number of centers suffice to achieve $\Theta(1)$-approximation. In case the order is random, Moshkovitz (2019) presented an algorithm that takes only $O(\log n)$ centers, no matter what the dataset is, and proved that the pathological dataset presented earlier provides a matching lower bound. In case the order is arbitrary and there are no assumptions on the dataset, $\Omega(n)$ centers is required, as discussed earlier. In this work, we fill up the gap and show that if $\mathrm{OC}_k(X) = \mathrm{polylog}(n)$, then the number of centers is upper and lower bounded by $\mathrm{polylog}(n)$, which means that, up to a polynomial, we use the same number of centers as the random order case.

OC \ order	random	worst
arbitrary	$\Theta(\log n)$	$\Omega(n)$
$\mathrm{polylog}(n)$	$O(\log n)$	$\Theta(\mathrm{polylog}(n))$

Table 1: Number of centers needed to achieve $\Theta(1)$-approximation compared to opt_k with constant k. The size of the input is n. In light blue, the contribution of this work. See the text for more details.

1.2.1. SUMMARY CONTRIBUTIONS

The contributions of the paper are summarized as follows.

Complexity measure. We introduce a new measure, $\mathrm{OC}_k(X)$, to identify X's complexity when clustering it in the online no-substitution setting and *arbitrary* order. It helps to quantify the number of points in X needed to be taken as centers. The new measure is the longest sub-series in X such that any point is far from all points preceding it.

Algorithm. We design a new random algorithm in the online no-substitution setting. It uses, as a subroutine, an approximated clustering algorithm A for the offline setting. For each new point x_t, the algorithm uses A to cluster all points observed so far. Then it performs a slight modification to A's clustering by merging all clusters that are close to x_t. The new cluster that x_t is in, $x_t \in C_i^t$, will determine the probability of taking x_t as center. See more details in Section 4.

Provable guarantees. We prove that when running the new algorithm on data X with an α-approximation offline clustering algorithm, it takes $O(\mathrm{OC}_k(X)\log(n)k\log k)$ centers and is an $O(\alpha k^3)$-approximation. We show that the number of centers needed to achieve an α-approximation is at least $\Omega\left(\frac{\mathrm{OC}_k(X)}{k\log(n\alpha)}\right)$. Specifically, suppose $\mathrm{OC}_k(X) = \mathrm{polylog}(n)$ and k, α are constants. In that case, the number of centers is lower and upper bounded by $\mathrm{polylog}(n)$, which is, up to a polynomial, similar to the random order case.

Applications. We prove that if the data X is sampled from a "natural" distribution, such as a mixture of Gaussians, then the new measure, $\mathrm{OC}_k(X)$, is equal to $O(k^2\log(n))$. Together with our provable guarantees, this implies that for data generated from "natural" distributions, our new algorithm takes only $\mathrm{poly}(k\log(n))$ centers and is a $\mathrm{poly}(k)$-approximation.

1.3. Related work

Online no-substitution setting. Several works Liberty et al. (2016); Moshkovitz (2019); Hess and Sabato (2020) designed algorithms in the online no-substitution setting. The works Hess and Sabato (2020); Moshkovitz (2019) assumed the order is random, and Hess and Sabato (2020); Liberty et al. (2016) assumed the data or the aspect ratio is bounded. Both assumptions simplify the problem. In this paper, we explore the case where the order is arbitrary, and data is unbounded. This paper shows that the number of centers is determined by $d = \mathrm{OC}_k(X)$. If the aspect ratio is small, then so is d, which implies similar results to Liberty et al. (2016). However, notably, small d does not force the entire data to be bounded or have a small aspect ratio. Thus we can handle cases previous works could not.

Online facility location. Meyerson Meyerson (2001) introduced the online variant of facility location. Demands arrive one after another and are assigned to a facility. Throughout the run of the algorithm, a set of facilities F is maintained. Upon arrival of each demand p there is a choice between (1) instant cost, $d(p,\ell)$, to p's closest location $l \in F$ (2) open a new facility. Opening a facility is irreversible. The total cost is $|F| + \sum_p d(p,\ell)$. Several variants of this problem were investigated (e.g., Fotakis (2011); Lang (2018); Feldkord and Meyer auf der Heide (2018)). One of the main differences between the online facility location and the online no-substitution setting is the term that we try to minimize: either the sum of $|F| + \sum_p d(p,\ell)$ or both terms separately. This seemingly small difference has a significant effect. For example, suppose $\sum_p d(p,\ell)$ is of the order of n. The online facility location algorithm can take $|F| = n$, permitting the trivial solution of opening a facility at each location. On the other hand, in the no-substitution setting, $|F|$ should be small.

Streaming with limited memory. There is a vast literature on algorithms in the streaming setting where the memory is limited Muthukrishnan (2005); Aggarwal (2007). One line of works uses core-sets Har-Peled and Mazumdar (2004); Phillips (2016); Feldman (2020), where a weighted subset, S, of the current input points X_t is saved such that a k-means solution to S has similar cost as X_t. Another line of work saves a set of candidate centers in memory Guha et al. (2000); Charikar et al. (2003); Shindler et al. (2011); Guha et al. (2003); Ailon et al. (2009). A different line of work had assumptions on the data like well-separated clusters Braverman et al. (2011); Ackerman and Dasgupta (2014); Raghunathan et al. (2017). Notably, in the setting of streaming with limited memory, decisions can be revoked, unlike this paper's requirement. Enabling a decision change when new information presents itself allows the algorithm to take a smaller number of points as centers.

2. Preliminaries

Notation. For convenience, we will denote the optimal cost with k centers as $\mathcal{L}_k(X) = \mathcal{L}(opt_k(X))$ and the cost of centers C as $\mathcal{L}(X, C)$. It will be frequently useful to consider the costs associated with using a single cluster center. Because of this, we let $\mathcal{L}(X, x)$ denote $\mathcal{L}(X, \{x\})$ and $\mathcal{L}(X)$ denote $\mathcal{L}_1(X)$.

1-means clustering. The optimal center of a cluster is its mean. The following well-known lemma will prove useful in our analysis.

Lemma 5 (center-shifting lemma) *Let $X \subset \mathbb{R}^d$ be a finite set and let $\mu = \frac{1}{|X|}\sum_{x\in X} x$. For any $x \in \mathbb{R}^d$, we have $\mathcal{L}(X, x) = \mathcal{L}(X) + |X|d(x, \mu)^2$.*

3. Online clustering using (α, k)-sequences

In this section, we introduce (α, k)-sequences, which are the principle object from which $\mathrm{OC}_k(X)$ is constructed.

Definition 6 *Let $\alpha > 1$. An (α, k)-sequence is an ordered sequence of points $x_1, y_2, \ldots, x_m$ such that for $1 < j \leq m$,*

$$\min_{1\leq i\leq j-1} d(x_i, x_j) > \alpha \mathrm{diam}_{k-1}(\{x_1, x_2, \ldots, x_{j-1}\}).$$

An (α, k)-sequence can be thought of a "worst-case" sequence for an online clustering algorithm. For each successive point, the algorithm is strongly incentivized to select that point, since doing so would incur a large cost compared to the cost for any other point.

The exact value of α in Definition 6 is insignificant, it solely essential that $\alpha > 1$. Converting an (α, k)-sequence into a (β, k)-sequence for some $\beta \neq \alpha$ has only slight effect on the length of sequence. We formalize this in the following lemma, which will also play a key role in providing lower bounds on the number of centers an approximation online algorithm must choose. See section section 5 for the proof.

Lemma 7 *Suppose $1 < \alpha < \beta$. Let $x_1, x_2, \ldots, x_n$ be an (α, k)-sequence. Then there exists a sub-sequence of length at least $\left\lfloor \frac{n}{2k \log_\alpha \beta} \right\rfloor$ that is a (β, k)-sequence.*

Lemma 7 shows that we can construct (α, k)-sequences from each other for different values of α at a relatively small reduction in sequence length. Because of this, the complexity measure we suggest, OC_k, essentially fixes $\alpha = 2$. Throughout this paper, we express both upper and lower bounds on the number of centers an online algorithm chooses given input X through $\mathrm{OC}_k(X)$.

4. An Algorithm for Online Clustering

In this section, we present and analyze our new online algorithm. At a high level, at each time step t, the algorithm computes an offline k-clustering of the points received so far $x_1, x_2, \ldots, x_t$. It then maximally merges clusters containing x_t (without increasing the total cost by more than a constant factor), and finally chooses x_t with probability inversely proportional to final merged cluster containing it.

The intuition here is that points end up in small merged clusters are more likely to be difficult to cluster with other points, while points in large ones are more likely to be easily clustered with others. Correspondingly, points in small clusters are chosen with high probability while points in large clusters are chosen with small probability.

To reduce the number of centers chosen, the algorithm modifies the offline clustering of the points $x_1, \ldots, x_t$ by maximally combining clusters so that x_t is contained in as large a cluster as possible without increasing the total cost too much. After doing so, it then chooses x_t as outlined above.

Algorithm 2: Online clustering algorithm for arbitrary order
Input: A stream of points $X = \{x_1, x_2, \ldots, x_n\}$, a desired number of clusters k, an offline clustering algorithm A
Output: A set of cluster centers $C \subseteq X$
$C \leftarrow \{\}$
for $t \in \{1, 2, 3, \ldots, n\}$ **do**
  $\mathcal{C}_t \leftarrow A(\{x_1, x_2, \ldots, x_t\}, k)$
  $\mathcal{C}_t = \{(C_t^1, c_t^1), (C_t^2, c_t^2), \ldots, (C_t^k, c_t^k)\}$
  without loss of generality $d(c_t^1, x_t) \leq d(c_t^2, x_t) \leq \cdots \leq d(c_t^k, x_t)$
  without loss of generality c_t^i is the optimal cluster center for C_t^i
  $v_t \leftarrow \max(\{i : \sum_{j=1}^{i} |C_t^j| d(c_t^j, c_t^i)^2 \leq 100\mathcal{L}(C_t^1, C_t^2, \ldots, C_t^k)\})$
  $s_t \leftarrow \sum_{j=1}^{v_t} |C_t^j|$
  with probability $\frac{20k \log k}{s_t}$, $C = C \cup \{x_t\}$
end
return C

Let v_t be defined as in Algorithm 2. It will prove useful to think of the center $c_t^{v_t}$ as replacing the centers $\{c_t^1, c_t^2, \ldots, c_t^{v_t}\}$. In essence, Algorithm 2 clusters $C_t^1, C_t^2, \ldots, C_t^{v_t}$ with center $c_t^{v_t}$, and clusters C_t^j with c_t^j for $j > v_t$. To this end, we let $c_t(x)$ denote the center that x is clustered with at time t. In particular, for any $x \in \{x_1, x_2, \ldots x_t\}$, let

$$c_t(x) = \begin{cases} c_t^{v_t} & x \in C_t^j, j \leq v_t \\ c_t^j & x \in C_t^j, j > v_t \end{cases}.$$

In the following lemma, we show that this "new" clustering still keeps the total cost at each time t bounded. That is, the new clustering used by Algorithm 2 is an $O(1)$-approximation to the optimal k-clustering at every time t.

Lemma 8 *For any* $1 \leq t \leq n$, $\sum_{i=1}^{t} d(x_i, c_t(x_i))^2 \leq 101\alpha\mathcal{L}_k(\{x_1, x_2, \ldots, x_t\})$.

Proof Observe that for $j \leq v_t$, we have $\sum_{x_i \in C_t^j} d(x_i, c_t(x_i))^2 = \mathcal{L}(C_t^j, c_t^{v_t})$, while for $j > v_t$, we have $\sum_{x_i \in C_t^j} d(x_i, c_t(x_i))^2 = \mathcal{L}(C_t^j, c_t^j) = \mathcal{L}(C_t^j)$. Summing these equations over all j and applying the center-shifting lemma, we have

$$\begin{aligned}
\sum_{i=1}^{t} d(x_i, c_t(x_i))^2 &= \sum_{j=1}^{k} \sum_{x_i \in C_t^j} d(x_i, c_t(x_i))^2 \\
&= \sum_{j=1}^{v_t} \mathcal{L}(C_t^j, c_t^{v_t}) + \sum_{j=v_t+1}^{k} \mathcal{L}(C_t^j) \\
&= \sum_{j=1}^{v_t} \mathcal{L}(C_t^j) + |C_t^j| d(c_t^j, c_t^{v_t})^2 + \sum_{j=v_t+1}^{k} \mathcal{L}(C_t^j) \\
&= \mathcal{L}(\mathcal{C}_t) + \sum_{j=1}^{v_t} |C_t^j| d(c_t^j, c_t^{v_t})^2.
\end{aligned}$$

By the definition of v_t, we have that $\sum_{j=1}^{v_t} |C_t^j| d(c_t^j, c_t^{v_t})^2 \leq 100\mathcal{L}(\mathcal{C}_t)$. Substituting this, implies that $\sum_{i=1}^{t} d(x_i, c_t(x_i))^2 \leq 101\mathcal{L}(\mathcal{C}_t)$. Since A is an approximation algorithm with approximation factor α, we have that $\mathcal{L}(\mathcal{C}_t) \leq \mathcal{L}_k(\{x_1, x_2, \ldots, x_t\})$, which implies the lemma. ■

In section 4.1, we show that this algorithm is an online $\mathrm{poly}(k)$-approximation algorithm, and in section 4.2, we analyze the expected number of centers chosen.

Theorem 23 implies that although there exist input sequences X for which any online approximation algorithm must take many centers (i.e. $\Omega(n)$), for input sequences X that are sampled from some well-behaved probability distribution, it is possible to do substantially better *regardless of the order X is presented in*.

4.1. Approximation factor analysis

The analysis of our algorithm hinges around the following known observation: taking a point at random from a single cluster yields a decent approximation for a cluster center. For completeness, we formalize this observation with the following lemma.

Definition 9 *Let S be any set of points, and let $G \subset S$ be defined as $G = \{x : \mathcal{L}(S, x) \leq 3\mathcal{L}(S)\}$. We refer to the points $g \in G$ as* good *points.*

Lemma 10 *Let S be any set of points, and let $G \subset S$ be the good points in S. Then $|G| \geq \frac{|S|}{2}$.*

Proof Let $n = |S|$ and $\mu = \frac{1}{|S|}\sum_{s\in S} s$ denote the mean of S. For $x \notin G$, $\mathcal{L}(S, x) > 3\mathcal{L}(S)$. By Lemma 5, $\mathcal{L}(S, x) = \mathcal{L}(S) + nd(x, \mu)^2$. Therefore $d(x, \mu)^2 > \frac{2\mathcal{L}(S)}{n}$. However, $\sum_{x\in S} d(x, \mu)^2 = \mathcal{L}(S)$ by definition. Thus we have

$$\mathcal{L}(S) \geq \sum_{x\notin G} d(x, \mu)^2 > \frac{2\mathcal{L}(S)|S \setminus G|}{n}.$$

This implies $\frac{|S\setminus G|}{n} < \frac{1}{2}$, which means $|G| > \frac{n}{2}$, as desired. ■

Lemma 10 implies that if points from S are independently selected with probability $\Theta\left(\frac{1}{|S|}\right)$, then it is likely that some point $g \in G$ will be selected. We will use this idea to argue that Algorithm 2 selects good points from each cluster in $opt_k(S)$ with high probability.

Theorem 11 *Let A be an offline clustering algorithm with approximation factor α. Suppose running Algorithm 2 on X, k, A returns a set of centers C. Then with probability 0.9 over the randomness of Algorithm 2,*

$$\mathcal{L}(X, C) \leq 1358\alpha k^3 \mathcal{L}_k(X).$$

Before giving a proof, we first describe our proof strategy and give some helpful definitions and lemmas.

Let $C_*^1, C_*^2, \ldots, C_*^k$ denote the optimal k-clustering of X, and let $G^1, G^2, \ldots, G^k$ denote the sets of good points in each cluster. We also let $|X| = n$.

Our proof strategy is the following. We will first show that there exist clusters C_*^i for which we are likely to choose some center $g \in G^i$. Therefore, for these clusters, we have a 3-approximation of the optimal cost. For the remaining clusters, we will argue that clusters that are not likely to have a good point chosen must be "close" to some other cluster. We will then conclude that the good points that we have already selected will serve as an approximation for *all* cluster C_*^i, which implies that the total cost is bounded by some constant times $\mathcal{L}_k(X)$.

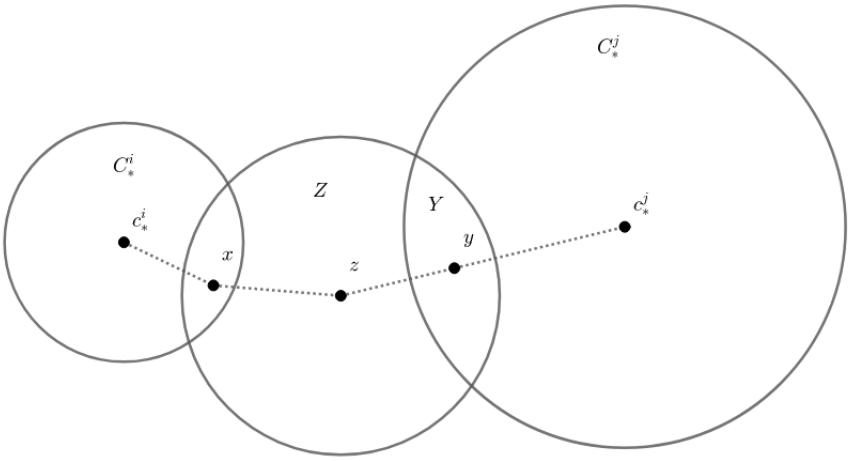
We begin with the first step. Using the notation from Algorithm 2, we let $s_t = |\cup_{i=1}^{v_t} C_t^i|$. For cases in which we don't explicitly note the index t, we will let $s(x)$ denote the same thing (i.e. $s(x_t) = s_t$). Here, $s(x)$ can be thought of as the set of points clustered with x (or near x).

Lemma 12 *Fix any $1 \leq i \leq k$. Suppose that for at least half the points $g \in G^i$, $s(g) \leq k|C_*^i|$. Then with probability at least $1 - \frac{1}{k^5}$, Algorithm 2 selects some $g \in G^i$.*

Proof For any g with $s(g) \leq k|C_*^i|$, Algorithm 2 selects g with probability at least $\frac{20k\log k}{k|C_*^i|} \geq \frac{10\log k}{|G^i|}$. Since there are at least $\frac{|G^i|}{2}$ such points g, and since each point is selected with an independent coin toss, Algorithm 2 selects *no* such points with probability at most $\left(1 - \frac{10\ln k}{|G^i|}\right)^{\frac{|G^i|}{2}}$. By standard manipulations, we have

$$\left(1 - \frac{10\ln k}{|G^i|}\right)^{\frac{|G^i|}{2}} \leq e^{-5\ln k} = \frac{1}{k^5},$$

as desired. ■



C_*^i: ith optimal cluster	c_*^i: center of C_*^i
C_*^j: jth optimal cluster	c_*^j: center of C_*^j
Z: $\cup_{j=1}^{v_t} C_t^i$	z: $c_t^{v_t}$
Y: $C_*^j \cap Z$	y: $\frac{1}{\lvert Y\rvert}\sum_{y' \in Y} y'$

Figure 1: Proof idea of Lemma 13. We derive upper bounds on all the dotted lines.

Next we do the second step, in which we handle clusters for which we are not likely to choose some good point $g \in G^i$.

Lemma 13 *Fix any $1 \leq i \leq k$. Suppose that for strictly less than half the points $g \in G^i$, $s(g) \leq k|C_*^i|$. Then there exists j with $|C_*^j| > |C_*^i|$ such that*

$$|C_*^i| d(c_*^i, c_*^j)^2 \leq 1254\alpha \mathcal{L}_k(X),$$

where c_*^i and c_*^j denote the centers of C_*^i and C_*^j respectively.

Proof Let t be the last time such that x_t is a good point from C_*^i (i.e. $x_t \in G^i$), and such that $s_t > k|C_*^i|$. By assumption, for strictly more than $\frac{|G^i|}{2}$ points x in G^i, $s(x) > k|C_*^i|$. This implies that $|\{x_1, x_2, \ldots, x_t\} \cap G^i| \geq \frac{|G^i|}{2}$. The idea is to analyze the algorithm at time t.

Let Z denote the combined cluster that x_t is assigned to at time t by Algorithm 2, that is $Z = \cup_{i=1}^{v_t} C_t^i$ and $z = c_{v_t}^t$. In particular, we have that $s_t = |Z|$. Because $|Z| > k|C_*^i|$, there exists $1 \leq j \leq k$ such that $|C_*^j \cap Z| > |C_*^i|$. We claim that for this value of j, C_*^j satisfies the desired properties in the lemma.

Our goal is to find an upper bound on $d(c_*^i, c_*^j)$. To do this, we will find bounds on $d(c_*^i, x_t)$, and $d(x_t, z)$, and $d(z, c_*^j)$, and then use the triangle inequality. For bounding $d(z, c_*^j)$ in particular, we will consider the intersection of Z and C_*^j which we denote as $Y = C_*^j \cap Z$. We let $y = \frac{1}{|Y|}\sum_{y' \in Y} y'$ be the average of all points in Y, and will subsequently bound $d(z, c_*^j)$ by bounding $d(z, y)$ and $d(y, c_j^*)$.

We will argue this by finding bounds on $d(c_*^i, x), d(x, z), d(z, y)$, and $d(y, c_*^j)$, and then using the triangle inequality. Figure 1 gives a picture that summarizes this argument. We will derive upper bounds on all the dotted lines.

Claim 1: $|C_*^i| d(c_*^i, x_t)^2 \le 2\mathcal{L}_k(X)$.

Since $x_t \in G^i$, we have $\mathcal{L}(C_*^i, x_t) \le 3\mathcal{L}(C_*^i)$. Therefore, we have $\mathcal{L}(C_*^i) + |C_*^i| d(c_*^i, x_t)^2 \le 3\mathcal{L}(C_i^*)$. Subtracting $\mathcal{L}(C_i^*)$ from both sides and substituting $\mathcal{L}(C_*^i) \le \mathcal{L}_k(X)$ gives the result.

Claim 2: $|C_*^i| d(x_t, z)^2 \le 526\alpha\mathcal{L}_k(X)$.

Let $G \subset G^i$ denote the set of all good points present at time t in C_*^i. In particular, we let $G = \{x_1, x_2, \ldots, x_t\} \cap G^i$. Recall that from the definition of t, we have $|G| \ge \frac{|G^i|}{2} \ge \frac{|C_*^i|}{4}$.

For any $x' \in G$ let z' be its closest center in $\{c_t^{v_t}, \ldots, c_t^k\}$. In particular, we have $z' = \arg\min_{c_t^j \in \{c_t^{v_t}, \ldots, c_t^k\}} d(x', c_t^j)$. The key observation is that x_t must be closer to z than it is to z' (from the definition of z). Applying the triangle inequality, it follows that $d(x_t, z) - d(x_t, x') \le d(x', z')$.

Observe that x_t and x' are both good points, and consequently by the argument in Claim 1, $|C_*^i| d(c_*^i, x_t)^2 \le 2\mathcal{L}_k(X)$ and $|C_*^i| d(c_*^i, x')^2 \le 2\mathcal{L}_k(X)$. Thus by applying the triangle inequality again, $|C_*^i| d(x_t, x')^2 \le 8\mathcal{L}_k(X)$.

Suppose that $d(x_t, z) \le d(x_t, x')$. Then this implies $|C_*^i| d(x_t, z)^2 \le 8\mathcal{L}_k(X)$ which implies the claim. In the other case, we assume $d(x_t, z) \ge d(x_t, x')$, which implies that $(d(x_t, z) - d(x_t, x'))^2 \le d(x', z')^2$. The idea now is to sum this equation over all $x' \in G$ and substitute $|C_*^i| d(x_t, x')^2 \le 8\mathcal{L}_k(X)$ to get that

$$|G| \left(d(x_t, z) - \sqrt{\frac{8\mathcal{L}_k(X)}{|C_*^i|}} \right)^2 \le \sum_{x' \in G} d(x', z')^2.$$

Since $G \subset \{x_1, x_2, \ldots, x_t\}$, it follows that $\sum_{x' \in G} d(x', z')^2$ is at most the cost of assigning each x_i to their nearest center in $\{c_t^{v_t}, c_t^{v_t+1}, \ldots, c_t^k\}$. This is upper bounded by Lemma 8, implying that $\sum_{x' \in G} d(x', z')^2 \le 101\alpha\mathcal{L}_k(X)$. Substituting this and observing that $|G| \ge \frac{|C_*^i|}{4}$, we have that

$$\begin{aligned}
|C_*^i| d(x_t, z)^2 &\le |C_*^i| \left(\sqrt{\frac{101\alpha\mathcal{L}_k(X)}{|G|}} + \sqrt{\frac{8\mathcal{L}_k(X)}{|C_*^i|}} \right)^2 \\
&\le |C_*^i| \left(\sqrt{\frac{404\alpha\mathcal{L}_k(X)}{|C_*^i|}} + \sqrt{\frac{8\mathcal{L}_k(X)}{|C_*^i|}} \right)^2 \\
&\le \left(\sqrt{404\alpha\mathcal{L}_k(X)} + \sqrt{8\mathcal{L}_k(X)} \right)^2 \\
&\le 526\alpha\mathcal{L}_k(X),
\end{aligned}$$

as desired.

Claim 3: $|C_*^i| d(z, y)^2 \le 101\alpha\mathcal{L}_k(X)$.

Observe that the cost incurred by Y at time t by the modified offline clustering (with centers $\{c_t^{v_t}, c_t^{v_t+1}, \ldots, c_t^k\}$) is $\mathcal{L}(Y, z) = \mathcal{L}(Y) + |Y| d(z, y)^2$. By Lemma 8, this cost is at most $101\alpha\mathcal{L}_k(X)$, and since $|Y| > |C_*^i|$, the result follows.

Claim 4: $|C_*^i| d(y, c_*^j)^2 \le \mathcal{L}_k(X)$.

Since $Y \subseteq C_*^j$, the cost $\mathcal{L}(Y, c_*^j)$ at time t by the modified offline clustering is at most $\mathcal{L}(C_*^j)$. Because $|Y| > |C_*^i|$, we have that $|C_*^i| d(y, c_*^j)^2 + \mathcal{L}(Y) \le \mathcal{L}(C_*^j) \le \mathcal{L}_k(X)$, which implies the result.

Putting it all together. Armed with all 4 of our claims, we can prove the lemma using the triangle inequality,

$$\begin{aligned}
|C_*^i| d(c_*^i, c_*^j)^2 &\le |C_*^i| (d(c_*^i, x_t) + d(x_t, z) + d(z, y) + d(y, c_*^j))^2 \\
&\le |C_*^i| \left(\sqrt{\frac{2\mathcal{L}_k(X)}{|C_*^i|}} + \sqrt{\frac{526\alpha\mathcal{L}_k(X)}{|C_*^i|}} + \sqrt{\frac{101\alpha\mathcal{L}_k(X)}{|C_*^i|}} + \sqrt{\frac{\mathcal{L}_k(X)}{|C_*^i|}} \right)^2 \\
&\le (\sqrt{2\alpha} + \sqrt{526\alpha} + \sqrt{101\alpha} + \sqrt{\alpha})^2 \mathcal{L}_k(X) \\
&\le 1254\alpha\mathcal{L}_k(X).
\end{aligned}$$

■

We now complete the proof of Theorem 11.

Proof (Theorem 11) Let T denote the set of all i such that for at least half the points $g \in G^i$, $s(g) \le k|C_*^i|$. Although T is a random set, its randomness only stems from the randomness in the approximation algorithm A. Crucially, the set T is independent from the results of the random choices (i.e. the last line of the for loop in Algorithm 2) that ultimately determine which elements of X we select in C. Thus, in this proof we will treat T and $s(g)$ as fixed entities, and evaluate all probabilities over the randomness from the random choices.

Using a union bound along with Lemma 12, we see that with probability at least $1 - \frac{k}{k^5} \ge 0.9$, we will choose some $g \in G^i$ for all $i \in T$. Recall that C denotes the output of Algorithm 2. Therefore, with probability 0.9, for all $i \in T$, we have

$$\mathcal{L}(C_*^i, C) \le 3\mathcal{L}(C_*^i).$$

Next, select any $1 \le i \le k$ with $i \notin T$. Although applying lemma 13 may not result in $j \in T$, it will result in j such that $|C_*^j| > |C_*^i|$. Therefore, applying this lemma at most k times will continually result in increasingly large sets $|C_*^j|$. Since such a C_*^j is guaranteed to exist, this must terminate in some $j \in T$.

Therefore, using the triangle inequality in conjunction with Cauchy Schwarz we see that there exists $j \in T$ such that

$$|C_i^*| d(c_*^i, c_*^j)^2 \le 1254\alpha k^2 \mathcal{L}_k(X).$$

Let g be a good point with $g \in G^j$. By the same argument given in Claim 1, $|C_*^i| d(g, c_*^j)^2 \le |C_*^j| d(g, c_*^j)^2 \le 2\mathcal{L}_k(X)$. Therefore, by the triangle inequality

$$|C_*^i| d(c_*^i, g)^2 \le |C_*^i| \left(d(c_*^i, c_*^j) + d(c_*^j, g) \right)^2 \le 1357\alpha k^2 \mathcal{L}_k(X).$$

By Lemma 5, this implies

$$\mathcal{L}(C_*^i, g) = |C_i^*| d(c_*^i, g)^2 + \mathcal{L}(C_*^i) \le 1358\alpha k^2 \mathcal{L}_k(X).$$

As shown earlier, Algorithm 2 selects some $g \in G^j$ with probability at least 0.9 for all $j \in T$. Therefore, with probability 0.9, Algorithm 2 outputs C such that

$$\mathcal{L}(X, C) \le \sum_{i=1}^{k} \mathcal{L}(C_*^i, C) \le \sum_{i=1}^{k} 1358\alpha k^2 \mathcal{L}_k(X) = 1358\alpha k^3 \mathcal{L}_k(X)$$

as desired.

■

4.2. Center complexity analysis

We now bound the expected number of centers outputted by Algorithm 2.

Theorem 14 *Let A be an offline α-approximation algorithm. If Algorithm 2 has output C, then*

$$\mathbb{E}[|C|] \leq 160k \log k \boldsymbol{OC}_k(X)(\log n + 1).$$

Before proving Theorem 14, we introduce some useful definitions and lemmas.

As before, we let $X = \{x_1, x_2, \ldots, x_n\}$ denote our input, and let s_t be as defined in Algorithm 2. Observe that x_i is selected as a center by our algorithm with probability $\frac{20k \log k}{s_i}$. Therefore, the expected number of centers satisfies

$$\mathbb{E}[|C|] = \sum_{i=1}^{n} \frac{20k \log k}{s_i}.$$

Our goal will be to show an upper bound on this expression. To do so, we will need the following constructions. For any $1 \leq t \leq n$, define the following.

1. Let $P_t \subseteq \{x_1, x_2, \ldots, x_t\}$ denote $P_t = C_t^1 \cup C_t^2 \cdots \cup C_t^{v_t}$. This represents all elements that were clustered with x_t by Algorithm 1 at time t.
2. For any x_t, we let $r_t = r(x_t) = d(x, c_t^{v_t+1})$. Thus $r(x_t)$ represents the distance from x_t to the closest center at time t that is not used for clustering P_t.
3. Let Q_t denote the set of all $x \in \{x_1, x_2, \ldots x_t\} \setminus P_t$ such that $d(x, c_t(x)) \geq \frac{1}{5} r(x_t)$, where $c_t(x)$ denote the cluster center c_j^t that x is assigned to at time t by offline clustering algorithm A.

Finally, we will use these sets P_t, Q_t for $1 \leq t \leq n$ to construct one directed graph G as follows. Let G have vertex set $\{1, 2, 3, \ldots, n\}$ and every vertex j have edges pointed to all i for which $x_i \in P_j$ or $x_i \in Q_j$. In particular, the set of edges in G denoted $E(G)$ satisfies

$$E(G) = \{(j, i) : x_i \in (P_j \cup Q_j)\}.$$

Our strategy will be show the following:

1. Any independent set $I \subset G$ forms a $(2, k)$-sequence.
2. G has an independent set of size at least $\frac{1}{8(\log n+1)} \sum_{i=1}^{n} \frac{1}{s_i}$.

These two observations will imply that

$$\mathbb{E}[|C|] = \sum_{i=1}^{n} \frac{20k \log k}{s_i} \leq 160k \log k \mathbf{OC}_k(X)(\log n + 1),$$

which is the desired result. We now verify these observations with the following lemmas.

Lemma 15 *If $i_1 \leq i_2, \cdots \leq i_r$ is an independent set in G, then $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ form a $(2, k)$-sequence.*

Proof Fix any $1 < s \leq r$, and for convenience let $t = i_s$ and let $I_s = \{x_{i_j} : 1 \leq j < s\}$ denote the set of points before x_{i_s}. The key observation is that $C_t^{v_t+1}, C_t^{v_t+2}, \ldots, C_t^k$ partition I_s into at most $k-1$ sets. For any $x_{i_j} \in I_s$, $x_{i_j} \notin P_t \cup Q_t$. Therefore, $d(x_{i_j}, c(x_{i_j})) < \frac{1}{5}r(x_t)$.

This means we have a partitioning of I_s into $k-1$ sets each of which has diameter strictly less than $\frac{2}{5}r(x_t)$. Meanwhile the distance from x_t to the closest point in I_s is at least $r(x_t) - \frac{1}{5}r(x_t) = \frac{4}{5}r(x_t)$. This is strictly more than double the $(k-1)$-diameter of I_s. Since s was arbitrary, we see that the precise condition for a $(2,k)$-sequence holds as desired. ■

Lemma 16 *For any t, $|P_t \cup Q_t| \leq 2s_t - 1$. Thus the vertex t has out-degree at most $2s_t - 1$ in G.*

Proof Recall that s_t was defined as $s_t = |C_t^1 \cup C_t^2 \cup \cdots \cup C_t^{v_t}| = |P_t|$. Therefore, it suffices to show that $|Q_t| \leq s_t - 1$.

Assume towards a contradiction that $|Q_t| \geq s_t$. Let L_t denote the cost of the original clustering chosen by Algorithm A at time step t. That is $L_t = \sum_{i=1}^k \mathcal{L}(C_t^i)$. By definition, each element in Q_t incurs a cost of at least $\frac{1}{25}r_t^2$. Therefore, we have that $L_t \geq \frac{s_t r_t^2}{25}$.

Next, we bound the cost of assigning $C_t^1, C_t^2, \ldots, C_t^{v_t}, C_t^{v_t+1}$ to $c_t^{v_t+1}$. To do so, observe that $d(x_t, c_t^i) \leq r_t$ for any $i \leq v_t + 1$. This is because the centers c_t^i are arranged in increasing order by their distance from x_t. Therefore, by the triangle inequality, for any $1 \leq i \leq v_t$, $d(c_t^i, c_t^{v_t+1}) \leq 2r_t$. This implies that

$$\begin{aligned}\sum_{i=1}^{v_t} |C_t^i| d(c_t^i, c_t^{v_t+1})^2 &\leq \sum_{i=1}^{v_t} |C_t^i| 4r_t^2 \\ &\leq 4s_t r_t^2 \\ &\leq 100 L_t,\end{aligned}$$

with the last inequality holding because $L_t \geq \frac{s_t r_t^2}{25}$. However, this contradicts the maximality of v_t, which implies that our assumption was false and $|Q_t| < s_t$ as desired. ■

Next, we will find a lower bound on the size of the largest independent set in G. Our main tool for doing so is Turan's theorem which we review in the following theorem. For completeness, we also include a proof.

Theorem 17 (Turan's theorem) *Let H be an undirected graph with average degree Δ. Then there exists an independent set in H consisting of at least $\frac{|H|}{\Delta+1}$ vertices, where $|H|$ denotes the number of vertices in H.*

Proof Let $|H| = m$ and let our vertices be labeled $1, 2, 3, \ldots m$. Let vertex i have degree Δ_i. Take a random ordering of the vertices in H. Proceed through the vertices in this order and select a vertex if and only if none of its neighbors has already been selected. At the end of this process, we are clearly left with an independent set I. The probability that vertex i is included in I is precisely $\frac{1}{\Delta_i+1}$, since i will be chosen if and only if it appears before its Δ_i neighbors. Thus, by linearity of expectation, we have that $\mathbb{E}[|I|] = \sum_{i=1}^m \frac{1}{\Delta_i+1}$. Thus there exists an independent set I^* with size at least $\sum_{i=1}^m \frac{1}{\Delta_i+1}$.

To finish the proof, let $f(x) = \frac{1}{x}$, thus $|I^*| \geq \sum_{i=1}^{m} f(\Delta_i + 1)$. The key observation is that f is a convex function on the interval $(0, \infty)$, and thus by Jensen's inequality, we have that

$$|I^*| \geq mf\left(\frac{\sum_{i=1}^{m} \Delta_i + 1}{m}\right) = mf(\Delta + 1) = \frac{m}{\Delta + 1},$$

as desired. ■

Lemma 18 *G has an independent set of size at least $\frac{1}{8(\log n+1)} \sum_{i=1}^{n} \frac{1}{s_i}$.*

Proof Let d_m denote the outdegree of vertex m. Partition the vertices of G, $\{1, 2, 3, \ldots, n\}$ into sets $S_0, S_1, S_2, \ldots, S_{\log n}$ such that

$$S_i = \{j : 2^{i-1} \leq d_j < 2^i\}.$$

We let S_0 be the set of all vertices with degree 0. Let G_i denote the subgraph of G induced by S_i. The main idea is to use Turan's theorem on each graph G_i, and then use the fact that there are $\log n$ graphs G_i to consider.

Observe that G_i has at most $|S_i|(2^i - 1)$ edges. By considering the undirected version of G (simply drop the orientation of each edge), it follows that the average degree is at most $\frac{|S_i|(2^{i+1}-2)}{|S_i|} = 2^{i+1} - 2$. Therefore, by Turan's theorem, G_i has an independent set I_i of size at least $\frac{|S_i|}{2^{i+1}}$. As a result, we have that

$$\sum_{j \in S_i} \frac{1}{d_j + 1} \leq \frac{|S_i|}{2^{i-1}} \leq 4|I_i|.$$

Let I denote the largest independent set of G. It follows that $I \geq |I_i|$ for all i. Summing the above inequality over all i, we see that

$$\sum_{j=1}^{n} \frac{1}{d_j + 1} \leq 4 \sum_{i=0}^{\log n} |I_i| \leq 4|I|(\log n + 1).$$

By Lemma 16, $d_m \leq 2s_m - 1$. Upon substituting this, the desired result follows. ■

We are now ready to prove Theorem 14.

Proof (Theorem 14) Let I be the largest independent set of G. By Lemma 15, we have that $|I| \leq \text{OC}_k(X)$. By Lemma 18, we have that

$$\frac{1}{8(\log n + 1)} \sum_{i=1}^{n} \frac{1}{s_i} \leq |I| \leq \text{OC}_k(X).$$

Multiplying by $\log n$, we see that

$$\mathbb{E}[|C|] = \sum_{i=1}^{n} \frac{20k \log k}{s_i} \leq 160k \log k \text{OC}_k(X)(\log n + 1),$$

as desired. ■

5. Lower Bounds

In this section, we prove lower bounds on the number of centers any online algorithm with approximation factor α must take. We first express these bounds in terms of $(\frac{1}{2}\sqrt{n\alpha}, k)$-sequences, and then convert them to bounds involving $\mathrm{OC}_k(X)$ by utilizing Lemma 7. The basic idea is that in an $(\frac{1}{2}\sqrt{n\alpha}, k)$ sequence, the points spread out at an extremely quickly rate. Therefore, each subsequent point must be selected for otherwise it incurs are large cost.

Proof [of Lemma 7] For any $1 < m \leq n$, let d_m denote the distance from x_m to the closest point preceding it, that is, $d_m = \min_{1 \leq i \leq m-1} d(x_i, x_m)$.

First, we claim that for any $m > k$, there exists $1 \leq i \leq k-1$ such that $d_m > \alpha d_{m-i}$. To see this, observe that by the definition of a (α, k)-sequence, it is possible to partition $\{x_1, x_2, \ldots, x_{m-1}\}$ into $k-1$ sets so that each has diameter strictly less than d_m/α. By the pigeonhole principle, at least one of $x_{m-1}, x_{m-2}, \ldots, x_{m-k+1}$ must be partitioned into a set with more than 1 element. Let $m-i$ be this value. Then, it follows that $d_m/\alpha > d_{m-i}$.

Next, by repeatedly applying this claim, starting with x_n, we can construct a sequence of points $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ such that $r \geq \frac{n}{k}$ and $d_{i_j} > \alpha d_{i_{j-1}}$ for all $1 < j \leq r$. Note that this sequence is constructed in reverse order by starting with $i_r = n$, and then setting $i_{r-1} = i_r - i$ where i is the value found by the argument above with $1 \leq i \leq k-1$.

Finally, let $s = \lceil \log_\alpha \beta \rceil$. It follows that for all j, $d_{i_j} > \alpha^{s-1} d_{i_{j-s+1}} > \frac{\beta}{\alpha} d_{i_{j-s+1}}$. Using the definition of an (α, k)-sequence, we have that for any j,

$$\begin{aligned} d_{i_j} &> \frac{\beta}{\alpha} d_{i_{j-s+1}} \\ &> \frac{\beta}{\alpha} \alpha \mathrm{diam}_{k-1}(\{x_1, x_2, \ldots, x_{i_{j-s}}\}) \\ &\geq \beta \mathrm{diam}_{k-1}(\{x_{i_{j-s}}, x_{i_{j-2s}}, x_{i_{j-3s}}, \ldots\}). \end{aligned}$$

By repeatedly setting j to be multiples of s, we see that $x_{i_s}, x_{i_{2s}}, \ldots, x_{i_{\lfloor r/s \rfloor s}}$, is a (β, k)-sequence. Thus, all the remains is to bound its length. Substituting $r \geq n/k$, we have that $\lfloor \frac{r}{s} \rfloor \geq \lfloor \frac{n}{k \lceil \log_\alpha \beta \rceil} \rfloor$, which implies the result. ■

Lemma 19 *Let $x_1, x_2, \ldots, x_n$ be an $(\frac{1}{2}\sqrt{n\alpha}, k)$-sequence. Then the expected number of centers taken by **any** streaming algorithm that guarantees approximation factor α is at least $0.9n$.*

Proof Consider any $1 < m \leq n$. Let d_m denote the distance from x_m to the closest point preceding it; that is, $d_m = \min_{1 \leq i \leq m-1} d(x_i, x_m)$. The key observation is that if the algorithm doesn't choose x_m, then it must pay a cost of at least d_m^2 at time m (since x_m must be clustered with some cluster center in $\{x_1, x_2, \ldots, x_{m-1}\}$). Because n is not known in advance, any streaming Algorithm must ensure that the cost at all times t is relatively low. We will show that d_m^2 is large enough so that failing to pick it will incur a cost at time m that is too high.

Consider the following clustering of $\{x_1, x_2, \ldots, x_m\}$. Let x_m be its own cluster, and then cluster $\{x_1, x_2, \ldots, x_{m-1}\}$ into $k-1$ clusters each with diameter strictly less than $\frac{2d_m}{\sqrt{n\alpha}}$. This is possible because of the definition of an $(\frac{1}{2}\sqrt{n\alpha}, k)$-sequence. It follows that each point is clustered with radius at most $\frac{d_m}{\sqrt{n\alpha}}$ in this clustering, and thus the entire cost is strictly less than $\frac{d_m^2}{\alpha}$. As a result, it follows that the total cost is small, that is, $\mathcal{L}_k(\{x_1, x_2, \ldots, x_m\}) < \frac{d_m^2}{\alpha}$.

Thus if an algorithm has approximation factor of α, it must select x_m with probability at least 0.9 for all $1 \leq m \leq n$, since otherwise it incurs cost at least $d_m^2 > \alpha \mathcal{L}_k(\{x_1, x_2, \ldots, x_m\})$. While it is possible that centers chosen in the future may incur a smaller cost for x_m, because n is unknown we can simply have the streaming stop at this point. The result follows. ■

By combining Lemmas 7 and 19, we get a lower bound for the number of centers an online algorithm must select given a worst case ordering of a dataset X.

Theorem 20 *Let X be an arbitrary set of points. There exists an ordering of X such that a streaming algorithm with approximation factor α must select at least $0.9\lfloor \frac{OC_k(X)}{k\lceil \log_2 \frac{1}{2}\sqrt{n\alpha} \rceil} \rfloor$ points in expectation.*

6. Bounds on $\mathrm{OC}_k(X)$ for mixture distributions

In this section, we consider the case where the input data X is generated from some distribution $\mathcal{D}$ over $\mathbb{R}^d$. While each point $x_i \in X$ is independently sampled from $\mathcal{D}$, we make no assumptions about the order in which these points are presented to our algorithm. Furthermore, in the k-clustering setting, it is natural to assume that $\mathcal{D}$ is a mixture of k distributions $\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^k$ over $\mathbb{R}^d$ such that each $\mathcal{D}^i$ corresponds to an "intrinsic" cluster of $\mathcal{D}$. In particular, we will find bounds on $\mathrm{OC}_k(X)$ under the assumption that each $\mathcal{D}^i$ is a relatively well behaved distribution.

We begin by defining the *aspect ratio* of a set X, which will subsequently be used to bound $\mathrm{OC}_k(X)$.

Definition 21 *The aspect ratio of a set of points $S = \{s_1, s_2, \ldots, s_n\}$, denoted asp$(S)$, is the ratio between the distance of the farthest two points of S and the closest two points of S. That is,*

$$asp(S) = \frac{\max_{i \neq j} d(s_i, s_j)}{\min_{i \neq j} d(s_i, s_j)}.$$

Let X be a dataset drawn from $\mathcal{D}$. As shown in Lemma 7, any $(2, k)$ subsequence have points with distances that grow exponentially. Furthermore, by the pigeonhole principle, at least $1/k$ of the elements in any $(2, k)$ sequence must come from some distribution $\mathcal{D}^i$. It follows that we can relate $\mathrm{OC}_k(X)$ to the aspect ratio $\mathrm{asp}(X^i)$, where X^i denotes the points in X drawn from $\mathcal{D}^i$.

Lemma 22 *For any set of points X and any $k \geq 1$, if $OC_k(X) \geq 1$, then for some $1 \leq i \leq k$, asp$(X^i) \geq 2^{OC_k(X)/k^2}$.*

Proof Let $\{x_1, x_2, \ldots, x_{\mathrm{OC}_k(X)/k}\} \subset X$ be the largest $(2, k)$ sequence in X for which all x_j are drawn from some $\mathcal{D}^i$. Such a sequence must exist by the definition of $\mathrm{OC}_k(X)$ and by a simple pigeonhole argument.

For any $1 < m \leq \mathrm{OC}_k(X)$, let $d_m = \min_{1 \leq i \leq m-1} d(x_i, x_m)$. By the argument given in the proof of Lemma 7, for any $m > k$ there exists $1 \leq i \leq k-1$ such that $d_m > 2d_{m-i}$. Thus applying this argument $\mathrm{OC}_k(X)/k$ times, we see that $d_{\mathrm{OC}_k(X)} > 2^{\mathrm{OC}_k(X)/k^2} d_i$ for some $2 \leq i \leq k$. The result follows from the definition of the aspect ratio. ■

We will now show that for a broad class of distributions $\mathcal{D}$ over $\mathbb{R}^d$, namely those for which each $\mathcal{D}^i$ has finite variance and bounded probability density, that $\mathrm{asp}(X \sim \mathcal{D}^n) = O(n^3)$. This in turn will imply that $\mathrm{OC}_k(X) \leq O(k^2 \log n)$.

Theorem 23 *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d$ that is a mixture of distributions $\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^k$. Suppose there exist constants D, ρ such that the following hold:*

1. *For each $1 \leq i \leq k$, the expected squared distance from $x \sim \mathcal{D}^i$ to its mean is bounded. In particular,*
$$\mathbb{E}_{x \sim \mathcal{D}^i}[d(x, \mathbb{E}_{x' \sim \mathcal{D}}[x'])^2] \leq D,$$
for some $0 < D < \infty$.
2. *$\mathcal{D}$ (the entire mixture distribution) has probability density at most ρ for some $0 < \rho < \infty$.*

Then for $X \sim \mathcal{D}^n$, with probability at least $1 - \frac{2}{n}$, $OC_k(X) = O(k^2 \log n)$.

Theorem 23 is proved through the following lemmas.

Lemma 24 *(VC theory) Let $\mathcal{X}$ denote any probability distribution over $\mathbb{R}^d$. For any ball $B \subset \mathbb{R}^d$, let $\mathbb{E}[B]$ denote $\mathbb{P}_{x \sim \mathcal{X}}[x \in B]$. For any $0 < \delta < 1$, let $\alpha_n = \sqrt{\frac{4(d+2)\ln(16n/\delta)}{n}}$. Then with probability $1 - \delta$ over $S \sim \mathcal{X}^n$, for **all** balls $B \subset \mathbb{R}^d$,*
$$\frac{|S \cap B|}{n} \leq \mathbb{E}[B] + \alpha_n^2 + \alpha_n \sqrt{\mathbb{E}[B]}.$$

For a proof of Lemma 24, see Lemma 1 of Dasgupta et al. (2007).

Lemma 25 *Let $\mathcal{D}$ be as described in Theorem 23. Then for $X = \{x_1, x_2, \ldots, x_n\} \sim \mathcal{D}^n$, with probability at least $1 - \frac{1}{n}$, $\min_{i \neq j} d(x_i, x_j) = \Omega(\frac{1}{n})$.*

Proof Let $\alpha_n = \sqrt{\frac{4(d+2)\ln(16n^2)}{n}}$ (we are setting $\delta = 1/n$ in the notation from Lemma 24). Let $r > 0$ be arbitrary. Then by Lemma 24, for all balls of radius r, we have that with probability $1 - \frac{1}{n}$ over $X \sim \mathcal{D}^n$, $\frac{|X \cap B|}{n} \leq \mathbb{P}_{x \sim \mathcal{D}}[x \in B] + \alpha_n^2 + \alpha_n \sqrt{\mathbb{P}_{x \sim \mathcal{D}}[x \in B]}$. We can bound $\mathbb{P}_{x \sim \mathcal{D}}[x \in B]$ by integrating the probability density of $\mathcal{D}$ over B. In particular, if C_d denotes the volume of the unit ball in $\mathbb{R}^d$, we have that $\mathbb{P}_{x \sim \mathcal{D}}[x \in B] \leq \int_B \rho d\mu = \rho C_d r^d$ where ρ is the upper bound on the probability density of $\mathcal{D}$. Substituting this, we see that with probability at least $1 - \frac{1}{n}$, for all balls of radius r,
$$\frac{|X \cap B|}{n} \leq \rho C_d r^d + \alpha_n^2 + \alpha_n \sqrt{\rho C_d r^d}.$$
By setting $r = \Omega(\frac{1}{n})$, and observing that $\alpha_n^2 << 1/n$ as $n \to \infty$, we see that with probability at least $1 - \frac{1}{n}$, $|X \cap B| < 2$ for all balls of radius r. By the triangle inequality, this implies that $\min_{i \neq j} d(x_i, x_j) \geq 2r = \Omega(\frac{1}{n})$, as desired. ■

Lemma 26 *Let $\mathcal{D}$ be as described in Theorem 23, and $X = \{x_1, x_2, \ldots, x_n\} \sim \mathcal{D}^n$. Let X^i be set of points in X sampled from $\mathcal{D}^i$ (as described earlier). Then with probability at least $1 - \frac{1}{n}$, for all $1 \leq i \leq k$, $\max_{x_a \neq x_b \in X^i} d(x_a, x_b) \leq O(kn^2)$.*

Proof Fix any $1 \leq i \leq k$. By the triangle inequality,

$$\max_{x_a \neq x_b \in X^i} d(x_a, x_b) \leq 2 \max_{x_a \in X^i} d(x_a, \mu^i),$$

where $\mu^i = \mathbb{E}_{x \sim \mathcal{D}^i}[x]$. Also, $|X^i| \leq |X| = n$. Therefore, by markov's inequality, we see that $\mathbb{P}_{x \sim \mathcal{D}^i}[d(x, \mu)^2 > Dkn^2] \leq \frac{1}{kn^2}$. Therefore, by a union bound over all $x^a \in X^i$, with probability at least $1 - \frac{1}{nk}$, $d(x_a, \mu^i)^2 \leq Dkn^2$ for all $x^a \in X^i$. Taking a union bound over all $1 \leq i \leq k$, gives the desired result. ■

We are now in the configuration to prove Theorem 23.

Proof (Theorem 23) By Lemmas 25 and 26, we have that for all $1 \leq i \leq k$, with probability at least $1 - \frac{2}{n}$, $\text{asp}(X^i) \leq O(kn^3)$. By Lemma 22, we have that for some i, $\text{OC}_k(X^i) \leq k^2 \log(\text{asp}(X))$ which implies that $\text{OC}_k(X) \leq O(k^2 \log n)$ as desired. ■

As an immediate consequence of Theorem 23, we see that for mixtures of k Gaussians, as well as for mixtures of k uniform distributions, $\text{OC}_k(X)$ is $O(k^2 \log n)$.

7. Conclusion and open questions

We design a new k-means clustering algorithm in the online no-substitution setting, where importantly, points are received in arbitrary order. We introduce a new complexity measure, $\text{OC}_k(X)$, to bound the number of centers the algorithm returns. We show that the complexity of data generated from many mixture distributions is bounded by $\text{OC}_k(X) = O(k^2 \log n)$. We prove that the algorithm takes only $O(\text{OC}_k(X) \log(n) k \log(k))$ centers, and the algorithm is a $\text{poly}(k)$-approximation. We complement this result by proving a lower bound of $\Omega\left(\frac{\text{OC}_k(X)}{k \log(\alpha n)}\right)$ on the number of centers taken by any α-approximation algorithm.

An obvious direction for future work is to improve the algorithm's parameters or prove they are tight. We proved that our algorithm is $\text{poly}(k)$-approximation. Can it be improved to $\Theta(1)$-approximation? For constant k we bounded the number of centers, taken by our algorithm, by $O(\text{OC}_k(X) \log(n))$ and showed a lower bound of $\Omega(\text{OC}_k(X)/\log(n))$, for any $\Theta(1)$-approximation algorithm. There is a gap of $\text{polylog}(n)$ between our lower and upper bounds on the number of centers. An interesting future work would be to close this gap.

The new algorithm and complexity measure are suggested to handle the case the order of the data is arbitrary, but can they also help if the order is random? It is known, Moshkovitz (2019), that if the order is random, then $\Omega(\log n)$ centers are necessary. This lower bound was proved using a high OC complexity dataset, which is equal to n. Suppose the data's complexity is $\text{OC}_k(X) \ll n$. Can the number of centers taken by a $\text{poly}(k)$-approximation algorithm be dependent solely on $\text{OC}_k(X)$ and not on $\log(n)$?

Acknowledgments

We thank Kamalika Chaudhuri for posing the central question of this paper: whether it is possible to obtain better results for online k-means clustering with adversarial order under assumptions on the underlying dataset (i.e. drawn from a mixture of Gaussians). We also thank Sanjoy Dasgupta for several helpful discussions about our results and proofs. Finally, R.B. thanks NSF under CNS 1804829 for research support.

References

Margareta Ackerman and Sanjoy Dasgupta. Incremental clustering: The case for extra clusters. In *Advances in Neural Information Processing Systems*, pages 307–315, 2014.

Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. Adaptive sampling for k-means clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 15–28. Springer, 2009.

Charu C Aggarwal. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.

Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in neural information processing systems*, pages 10–18, 2009.

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Vladimir Braverman, Adam Meyerson, Rafail Ostrovsky, Alan Roytman, Michael Shindler, and Brian Tagiku. Streaming k-means on well-clusterable data. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 26–40. Society for Industrial and Applied Mathematics, 2011.

Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. Better streaming algorithms for clustering problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 30–39, 2003.

Sanjoy Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, 2008.

Sanjoy Dasgupta, Daniel J. Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 353–360. Curran Associates, Inc., 2007.

Björn Feldkord and Friedhelm Meyer auf der Heide. Online facility location with mobile facilities. In *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*, pages 373–381, 2018.

Dan Feldman. Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 23–44. Springer, 2020.

Dimitris Fotakis. Online and incremental algorithms for facility location. *ACM SIGACT News*, 42 (1):97–131, 2011.

Sudipto Guha, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams. In *The 41st Annual Symposium on Foundations of Computer Science*, 2000.

Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE transactions on knowledge and data engineering*, 15(3):515–528, 2003.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, 2004.

Tom Hess and Sivan Sabato. Sequential no-substitution k-median-clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 962–972, 2020.

Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

Harry Lang. Online facility location against at-bounded adversary. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1002–1014. SIAM, 2018.

Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. An algorithm for online k-means clustering. In *2016 Proceedings of the eighteenth workshop on algorithm engineering and experiments (ALENEX)*, pages 81–89. SIAM, 2016.

Beth Logan. Music recommendation from song sets. In *ISMIR*, pages 425–428, 2004.

Adam Meyerson. Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 426–431. IEEE, 2001.

Michal Moshkovitz. Unexpected effects of online k-means clustering. *arXiv preprint arXiv:1908.06818*, 2019.

Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

Jeff M Phillips. Coresets and sketches. *arXiv preprint arXiv:1601.00617*, 2016.

Aditi Raghunathan, Prateek Jain, and Ravishankar Krishnawamy. Learning mixture of gaussians with streaming data. In *Advances in Neural Information Processing Systems*, pages 6605–6614, 2017.

Andrei Sorin Sabau. Survey of clustering based financial fraud detection research. *Informatica Economica*, 16(1):110, 2012.

Michael Shindler, Alex Wong, and Adam W Meyerson. Fast and accurate k-means for large datasets. In *Advances in neural information processing systems*, pages 2375–2383, 2011.

Bichen Zheng, Sang Won Yoon, and Sarah S Lam. Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms. *Expert Systems with Applications*, 41(4):1476–1482, 2014.